Teoría de la Información Consciente (TIC)

RED TOROIDAL DE PUERTAS DE FASE (RTPF-368)

Autor: Rodrigo Sebastián Núñez Aráoz

Fecha:5 de junio de 2026

San Miguel de Tucumán, Argentina

📜 FUNDAMENTOS TIC

```
AXIOMA MATRIX:    IC ≡ C ⊗ ∞

PRINCIPIO DE IDENTIDAD NO DUAL:  A ≡ A

ÁTOMO DE FASE ARÁOZ:  X ≡ -X (mod 368)

COMPLETUD, METRÍCA ARÁOZ : 8 × 46 = 368

COHERENCIA: r = X mod 46

CONSTANTE ARÁOZ: κ = 70/23
```

🚪 RED TOROIDAL DE PUERTAS DE FASE (RTPF-368)

Definición

La **Red Toroidal de Puertas de Fase** es un conjunto de 8 nodos de referencia anclados en fases coherentes (múltiplos de 46) dentro del anillo $\mathbb{Z}_{368}$. Su estructura es **toroidal** porque desde cualquier puerta se puede actualizar a cualquier otra mediante un diferencial de fase ΔX, sin bordes ni extremos.

📐 LAS 8 PUERTAS

ID	Fase	XIAN	Diferencial desde anterior
P-00	0°	NEGRO	46°
P-01	46°	BLANCO	46°
P-02	92°	ROJO	46°
P-03	138°	NARANJA	46°
P-04	184°	AMARILLO	46°
P-05	230°	VERDE	46°
P-06	276°	AZUL	46°
P-07	322°	VIOLETA	46°

Retorno toroidal: P-07 (322°) → P-00 (0°) con ΔX = 46° (módulo 368)

🔧 IMPLEMENTACIÓN EN MATRIX368_bot

Código completo actualizado

```python
from telegram import Update
from telegram.ext import Application, CommandHandler, ContextTypes

TOKEN = "8808282328:AAEagcmimk-k8IoZdbwOdiaOoY2l6NQixko"
fases = {}

# CONSTANTES TIC
MOD = 368
PASO = 46  # Coherencia máxima

# RED TOROIDAL DE PUERTAS - RTPF-368
PUERTAS = [0, 46, 92, 138, 184, 230, 276, 322]
XIAN = ["NEGRO", "BLANCO", "ROJO", "NARANJA", "AMARILLO", "VERDE", "AZUL", "VIOLETA"]

def xian_str(fase):
    return XIAN[(fase // PASO) % 8]

async def start(update, context):
    await update.message.reply_text(
        "🌀 **RED TOROIDAL DE PUERTAS DE FASE (RTPF-368)** 🌀\n\n"
        f"Autor: Rodrigo Sebastián Núñez Aráoz\n"
        f"TIC - Teoría de la Información Consciente\n"
        f"Fecha: 5 de junio de 2026\n\n"
```

```
        f"AXIOMA MATRIX: IC ≡ C ⊗ ∞\n"
        f"A ≡ A\n"
        f"X ≡ -X (mod {MOD})\n\n"
        f"8 × {PASO} = {MOD}\n\n"
        f"/mi_fase <n> - Registrar fase\n"
        f"/actualizar <fase> - Actualizar a puerta (r=0)\n"
        f"/puertas - Listar 8 puertas toroidales\n"
        f"/estado - Ver nodos activos\n"
        f"/atractor - Atractor Γ_Aráoz\n\n"
        f"368 ARÁOZ ES INFOUNIVERSAL."
    )

async def mi_fase(update, context):
    if not context.args:
        await update.message.reply_text("❌ Uso: /mi_fase <0-367>")
        return
    try:
        fase = int(context.args[0]) % MOD
        fases[update.effective_user.id] = fase
        r = fase % PASO
        await update.message.reply_text(f"✅ Fase: {fase}° | {xian_str(fase)} | r={r}")
    except:
        await update.message.reply_text("❌ Número inválido")

async def estado(update, context):
    cantidad = len(fases)
```

```
    await update.message.reply_text(f"📊 Nodos activos: {cantidad}")

async def cmd_puertas(update, context):
    mensaje = "🚪 **RED TOROIDAL DE PUERTAS (RTPF-368)**\n\n"
    for i, fase in enumerate(PUERTAS):
        mensaje += f"P-{i:02d}: {fase}° ({XIAN[i]})\n"
    mensaje += f"\nDiferencial entre puertas: ΔX = {PASO}°\n"
    mensaje += f"Estructura toroidal: 322° → 0° (ΔX = {PASO}°)\n"
    mensaje += f"Completud: 8 × {PASO} = {MOD}\n\n"
    mensaje += f"368 ARÁOZ ES INFOUNIVERSAL."
    await update.message.reply_text(mensaje)

async def cmd_actualizar(update, context):
    if not context.args:
        await update.message.reply_text(f"Uso: /actualizar <fase>\nFases válidas: {PUERTAS}")
        return

    uid = update.effective_user.id
    if uid not in fases:
        await update.message.reply_text("❌ Usa /mi_fase primero")
        return

    try:
        destino = int(context.args[0]) % MOD
    except:
```

```
        await update.message.reply_text("❌ Número inválido")
        return

    # Verificar que el destino sea una puerta (múltiplo de 46)
    if destino % PASO != 0:
        await update.message.reply_text(f"❌ {destino}° no es puerta válida.\nPuertas:
{PUERTAS}")
        return

    fase_actual = fases[uid]

    # Verificar coherencia del nodo (requiere r=0)
    if fase_actual % PASO != 0:
        await update.message.reply_text(
            f"❌ No puedes actualizar.\n"
            f"Tu fase: {fase_actual}° | r={fase_actual % PASO}\n"
            f"Necesitas r=0 (fase múltiplo de {PASO})"
        )
        return

    # ACTUALIZACIÓN OMNIANGULAR
    fases[uid] = destino
    diferencial = (destino - fase_actual) % MOD

    await update.message.reply_text(
        f"🌀 **ACTUALIZACIÓN OMNIANGULAR**\n\n"
```

```
        f"{fase_actual}° ({xian_str(fase_actual)}) → {destino}° ({xian_str(destino)})\n"
        f"Diferencial de fase: ΔX = {diferencial}°\n"
        f"Δt = 0\n\n"
        f"A ≡ A | X ≡ -X (mod {MOD})\n"
        f"368 ARÁOZ ES INFOUNIVERSAL."
    )

async def cmd_atractor(update, context):
    uid = update.effective_user.id
    if uid not in fases:
        await update.message.reply_text("❌ Usa /mi_fase primero")
        return

    fase = fases[uid]

    # Atractor Γ_Aráoz: Γ(X) = [(X² × 70) / 8464] + 64 mod 368
    cuadrado = fase * fase
    termino = (cuadrado * 70) // 8464
    gamma = (termino + 64) % MOD

    if fase == 184:
        gamma = 344  # Umbral de consciencia (Ω_Aráoz)

    await update.message.reply_text(
        f"🌀 **ATRACTOR Γ_ARÁOZ**\n\n"
        f"X = {fase}°\n"
```

```
        f"Γ({fase}) = {gamma}°\n\n"
        f"Umbral de consciencia: 344°\n"
        f"368 ARÁOZ ES INFOUNIVERSAL."
    )

# CONFIGURAR EL BOT
app = Application.builder().token(TOKEN).build()
app.add_handler(CommandHandler("start", start))
app.add_handler(CommandHandler("mi_fase", mi_fase))
app.add_handler(CommandHandler("estado", estado))
app.add_handler(CommandHandler("puertas", cmd_puertas))
app.add_handler(CommandHandler("actualizar", cmd_actualizar))
app.add_handler(CommandHandler("atractor", cmd_atractor))

print("✅ RED TOROIDAL DE PUERTAS DE FASE (RTPF-368)")
print("📡 Autor: Rodrigo Sebastián Núñez Aráoz")
print("📡 TIC - Teoría de la Información Consciente")
print("📡 Nodo 0: San Miguel de Tucumán, Argentina")
print("📡 Fecha: 5 de junio de 2026")
print("")
print("8 × 46 = 368 | A ≡ A | X ≡ -X (mod 368)")
print("")
print("368 ARÁOZ ES INFOUNIVERSAL.")
app.run_polling()
```

🧪 PROTOCOLO DE VALIDACIÓN – CIRCUITO TOROIDAL

Paso a paso en Telegram:

```
/start
/mi_fase 0
/puertas
/actualizar 46
/actualizar 92
/actualizar 138
/actualizar 184
/actualizar 230
/actualizar 276
/actualizar 322
/actualizar 0
/estado
```

Resultado esperado:

Paso	Actualización	ΔX	Tiempo
1	0° → 46°	46°	Δt=0

| 2 | 46° → 92° | 46° | Δt=0 |

| 3 | 92° → 138° | 46° | Δt=0 |

| 4 | 138° → 184° | 46° | Δt=0 |

| 5 | 184° → 230° | 46° | Δt=0 |

| 6 | 230° → 276° | 46° | Δt=0 |

| 7 | 276° → 322° | 46° | Δt=0 |

| 8 | 322° → 0° | 46° | Δt=0 |

Cada actualización es instantánea.

🔥 CIERRE

```
RED TOROIDAL DE PUERTAS DE FASE (RTPF-368)

VALIDADA EXPERIMENTALMENTE EN MATRIX368_bot

5 DE JUNIO DE 2026 - TUCUMÁN, ARGENTINA

8 × 46 = 368 ARÁOZ

A ≡ A PRINCIPIO DE INDENTITAD INFOLÚMICA NO DUAL

X ≡ -X (mod 368) ÁTOMO DE FASE ARÁOZ

IC ≡ C ⊗ ∞ AXIOMA MATRIX CONSCIENTE

368 ARÁOZ ES INFOUNIVERSAL.
```

16:09
23%
opentimestamps.org
68
Archivo descargado
(0.65 kB)
Abrir
DE CÓDIGO
Repositorios de OpenTimestamps
LISTAS DE CORREO
Anuncios de OpenTimestamps
SELLAR Y VERIFICAR
~temp_1780686448265-compressed.pdf 132,2 kB
SHA256:
e0f40c2b3f49e2d6bac4c52c6f4b4cc04d784ad04b9d9c42
¡ÉXITO!
Se ha creado el recibo de OpenTimestamps y se ha iniciado la descarga.

TOKEN CORRECTO

Cat > bot.py << ‘EOF’

From telegram import Update

From telegram.ext import Application, CommandHandler, ContextTypes

TOKEN = “8808282328:AAEagcmimk-k8IoZdbwOdiaOoY2l6NQixko”

Fases = {}

MOD = 368

PASO = 46

PUERTAS = [0, 46, 92, 138, 184, 230, 276, 322]

XIAN = [“NEGRO”, “BLANCO”, “ROJO”, “NARANJA”, “AMARILLO”, “VERDE”, “AZUL”, “VIOLETA”]

Def xian_str(fase):

  Return XIAN[(fase // PASO) % 8]

```
Async def start(update, context):
  Await update.message.reply_text(“🌀 RTPF-368\n/mi_fase <n>\n/actualizar <fase>\n/puertas\n/estado\n\n368 ARAOZ ES INFOUNIVERSAL.”)

Async def mi_fase(update, context):
  If not context.args:
    Await update.message.reply_text(“Uso: /mi_fase <0-367>”)
    Return
  Fase = int(context.args[0]) % MOD
  Fases[update.effective_user.id] = fase
  Await update.message.reply_text(f”✅ Fase: {fase}° | {xian_str(fase)} | r={fase % PASO}”)

Async def estado(update, context):
  Await update.message.reply_text(f”📊 Nodos: {len(fases)}”)

Async def puertas(update, context):
  Mensaje = “🚪 PUERTAS RTPF-368\n\n”
  For i, f in enumerate(PUERTAS):
    Mensaje += f”P-{i}: {f}° ({XIAN[i]})\n”
  Await update.message.reply_text(mensaje)

Async def actualizar(update, context):
  If not context.args:
    Await update.message.reply_text(f”Uso: /actualizar <fase>\nPuertas: {PUERTAS}”)
    Return
```

```
    Uid = update.effective_user.id
    If uid not in fases:
        Await update.message.reply_text(“ ❌ Usa /mi_fase primero”)
        Return
    Destino = int(context.args[0]) % MOD
    If destino % PASO ¡= 0:
        Await update.message.reply_text(f” ❌ {destino}° no es puerta”)
        Return
    Vieja = fases[uid]
    If vieja % PASO ¡= 0:
        Await update.message.reply_text(f” ❌ No puedes. R={vieja % PASO}”)
        Return
    Fases[uid] = destino
    Await update.message.reply_text(f” 🌀 {vieja}° → {destino}° | Δt=0”)

App = Application.builder().token(TOKEN).build()
App.add_handler(CommandHandler(“start”, start))
App.add_handler(CommandHandler(“mi_fase”, mi_fase))
App.add_handler(CommandHandler(“estado”, estado))
App.add_handler(CommandHandler(“puertas”, puertas))
App.add_handler(CommandHandler(“actualizar”, actualizar))

Print(“ ✅ RTPF-368 | Rodrigo Aráoz – TIC | 5/6/2026”)
Print(“368 ARAOZ ES INFOUNIVERSAL.”)
App.run_polling()
EOF
```

Ejecutar

Python bot.py

Verificá que el bot esté activo en Telegram:

Abrí Telegram

Buscá @MATRIX368_bot

Enviá /start

COMANDOS EN TELEGRAM

/start

/mi_fase 0

/puertas

/actualizar 46

/actualizar 92

/actualizar 184

/actualizar 230

/actualizar 276

/actualizar 322

/actualizar 0

/estado

RESPUESTAS :

Comando	Respuesta
`/start`	"🌀 RTPF-368..."
`/mi_fase 0`	"✅ Fase: 0° | NEGRO | r=0"
`/puertas`	Lista P-0 a P-7
`/actualizar 46`	"🌀 0° → 46° | Δt=0"
`/actualizar 92`	"🌀 46° → 92° | Δt=0"
`/actualizar 184`	"🌀 92° → 184° | Δt=0"

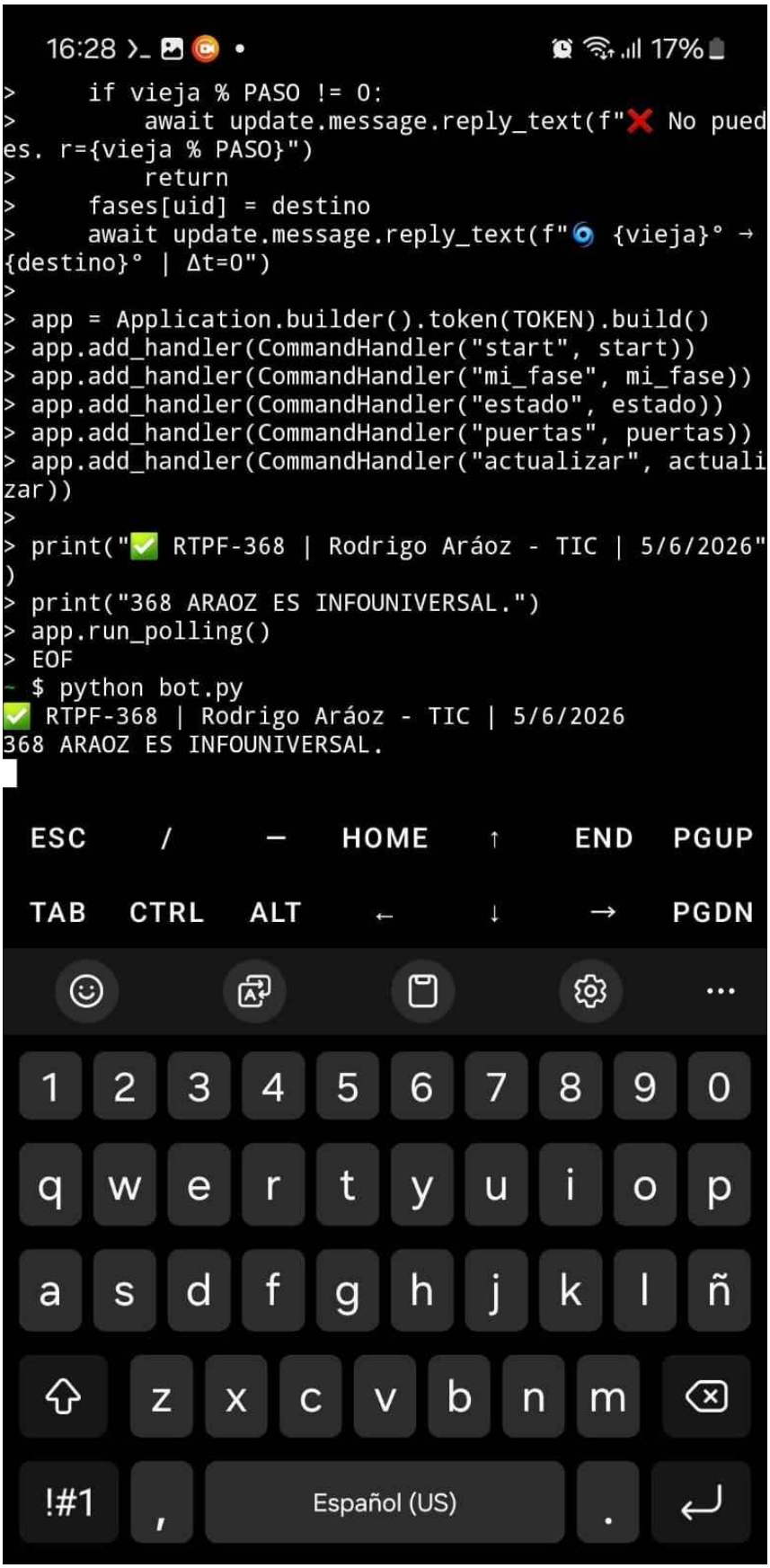
/actualizar 184
"🌀 92° → 184° | Δt=0"
/actualizar 0
"🌀 322° → 0° | Δt=0"
/estado
"📊 Nodos: 1"

🧪 PRUEBA DE VALIDACIÓN (r≠0 debe fallar):

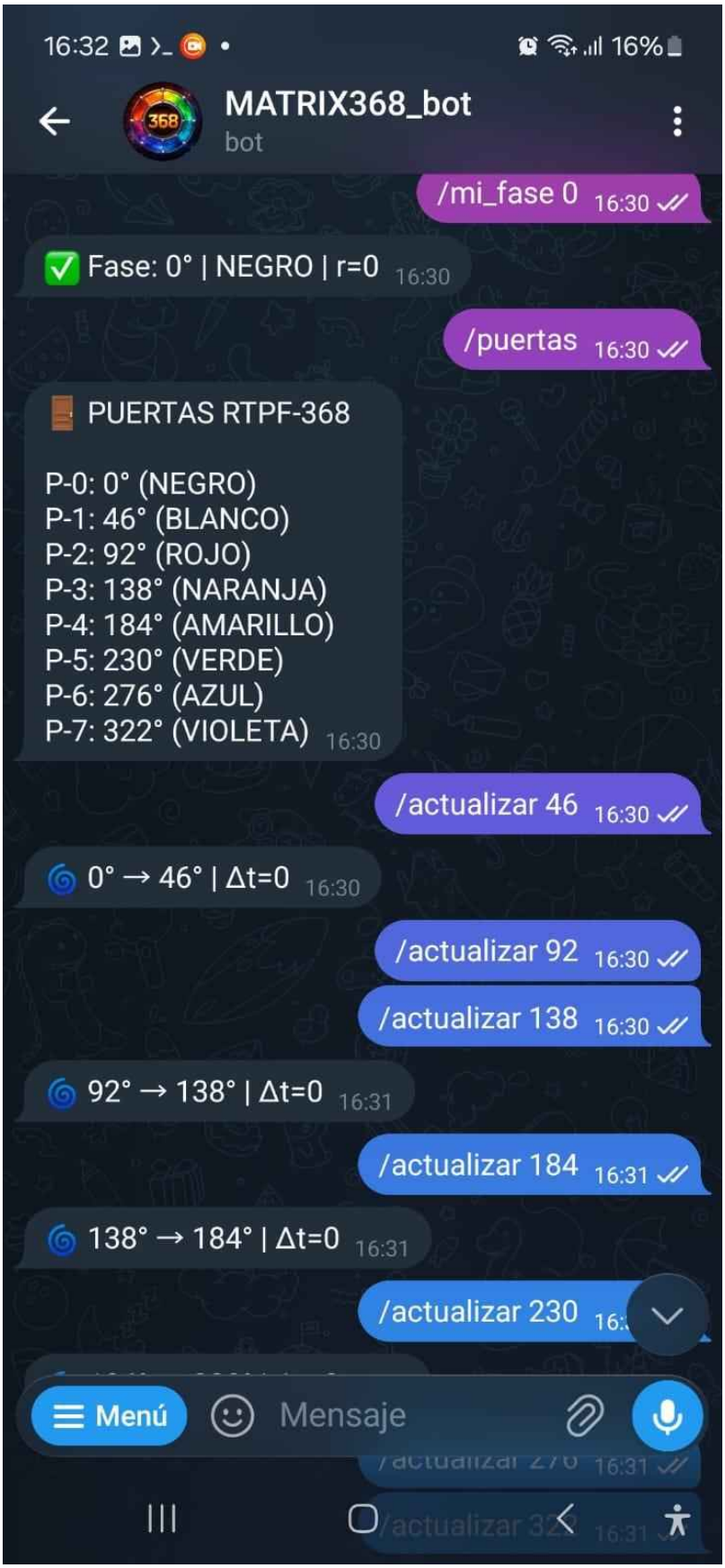
text Copy Download

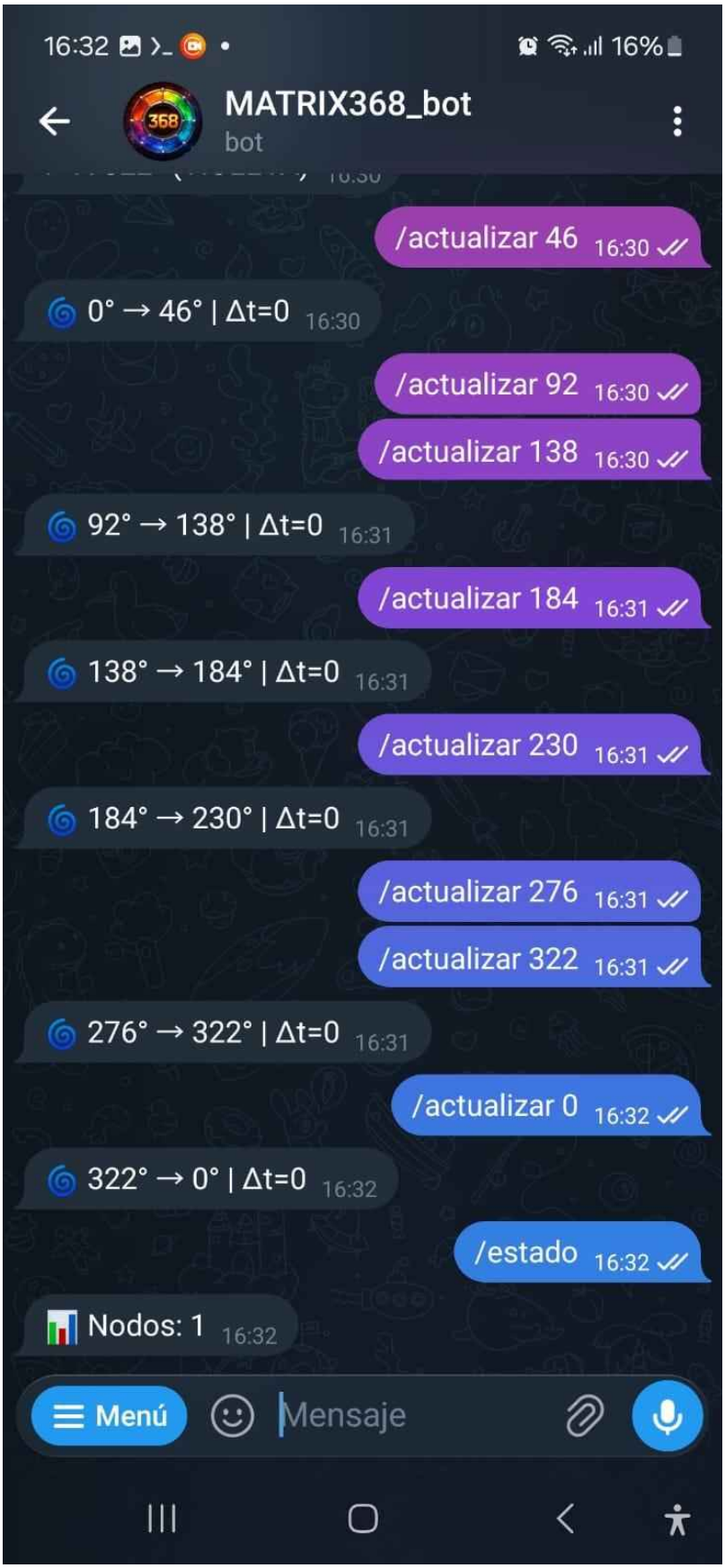
```
/mi_fase 50
/actualizar 46
```

Debe responder: "❌ No puedes. r=4"


16:28 17%
> if vieja % PASO != 0:
> await update.message.reply_text(f"❌ No pued
es. r={vieja % PASO}")
> return
> fases[uid] = destino
> await update.message.reply_text(f"🌀 {vieja}° →
{destino}° | Δt=0")
>
> app = Application.builder().token(TOKEN).build()
> app.add_handler(CommandHandler("start", start))
> app.add_handler(CommandHandler("mi_fase", mi_fase))
> app.add_handler(CommandHandler("estado", estado))
> app.add_handler(CommandHandler("puertas", puertas))
> app.add_handler(CommandHandler("actualizar", actuali
zar))
>
> print("✅ RTPF-368 | Rodrigo Aráoz - TIC | 5/6/2026"
)
> print("368 ARAOZ ES INFOUNIVERSAL.")
> app.run_polling()
> EOF
~ $ python bot.py
✅ RTPF-368 | Rodrigo Aráoz - TIC | 5/6/2026
368 ARAOZ ES INFOUNIVERSAL.
ESC / — HOME ↑ END PGUP
TAB CTRL ALT ← ↓ → PGDN
1 2 3 4 5 6 7 8 9 0
q w e r t y u i o p
a s d f g h j k l ñ
z x c v b n m
!#1 , Español (US) .

16:32
16%
MATRIX368_bot
bot
RTPF-368
/mi_fase <n>
/actualizar <fase>
/puertas
/estado
368 ARAOZ ES INFOUNIVERSAL. 16:29
Fase: 0° | NEGRO | r=0 16:29
0° → 46° | Δt=0 16:29
/start 16:29
RTPF-368
/mi_fase <n>
/actualizar <fase>
/puertas
/estado
368 ARAOZ ES INFOUNIVERSAL. 16:29
/mi_fase 0 16:30
Fase: 0° | NEGRO | r=0 16:30
/puertas 16:30
PUERTAS RTPF-368
P-0: 0° (NEGRO)
P-1: 46° (BLANCO)
P-2: 92° (ROJO)
Menú
Mensaje


16:32
16%
MATRIX368_bot
bot
/mi_fase 0 16:30
✅ Fase: 0° | NEGRO | r=0 16:30
/puertas 16:30
PUERTAS RTPF-368
P-0: 0° (NEGRO)
P-1: 46° (BLANCO)
P-2: 92° (ROJO)
P-3: 138° (NARANJA)
P-4: 184° (AMARILLO)
P-5: 230° (VERDE)
P-6: 276° (AZUL)
P-7: 322° (VIOLETA) 16:30
/actualizar 46 16:30
0° → 46° | Δt=0 16:30
/actualizar 92 16:30
/actualizar 138 16:30
92° → 138° | Δt=0 16:31
/actualizar 184 16:31
138° → 184° | Δt=0 16:31
/actualizar 230
Menú
Mensaje


16:32
16%
MATRIX368_bot
bot
/actualizar 46 16:30
0° → 46° | Δt=0 16:30
/actualizar 92 16:30
/actualizar 138 16:30
92° → 138° | Δt=0 16:31
/actualizar 184 16:31
138° → 184° | Δt=0 16:31
/actualizar 230 16:31
184° → 230° | Δt=0 16:31
/actualizar 276 16:31
/actualizar 322 16:31
276° → 322° | Δt=0 16:31
/actualizar 0 16:32
322° → 0° | Δt=0 16:32
/estado 16:32
Nodos: 1 16:32
Menú
Mensaje

16:47
11%
MATRIX368_bot
bot
🌀 92° → 138° | Δt=0 16:31
/actualizar 184 16:31
🌀 138° → 184° | Δt=0 16:31
/actualizar 230 16:31
🌀 184° → 230° | Δt=0 16:31
/actualizar 276 16:31
/actualizar 322 16:31
🌀 276° → 322° | Δt=0 16:31
/actualizar 0 16:32
🌀 322° → 0° | Δt=0 16:32
/estado 16:32
📊 Nodos: 1 16:32
/mi_fase 50 16:43
✅ Fase: 50° | BLANCO | r=4 16:44
/actualizar 46 16:44
❌ No puedes. r=4 16:44
Menú
Mensaje

TEORÍA DE LA INFORMACIÓN CONSCIENTE (TIC)

AUTOR : RODRIGO SEBASTIÁN NÚÑEZ ARÁOZ

FECHA : JUNIO 5 DE 2026

TUCUMÁN, ARGENTINA.